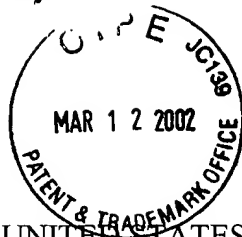


00862.022409.



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
TADAYOSHI NAKAYAMA) Examiner: Not Yet Assigned
Application No.: 09/982,916) Group Art Unit: 2181
Filed: October 22, 2001)
For: FILTER PROCESSING)
APPARATUS AND ITS)
CONTROL METHOD, PROGRAM)
AND STORAGE MEDIUM : March 11, 2002

Commissioner for Patents
Washington, D.C. 20231

CLAIM TO PRIORITY

RECEIVED
MAR 1 4 2002
Technology Center 2100

Sir:

Applicant hereby claims priority under the International Convention and all rights to which he is entitled under 35 U.S.C. § 119 based upon the following Japanese

Priority Applications:

2000-323040, filed October 23, 2000;

2000-344311, filed November 10, 2000; and

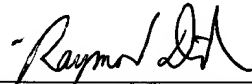
2000-399331, filed December 27, 2000.

Certified copies of the priority documents are enclosed.

Applicant's undersigned attorney may be reached in our New York office by

telephone at (212) 218-2100. All correspondence should continue to be directed to our address given below.

Respectfully submitted,



Attorney for Applicant

Registration No. 44,063

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3801
Facsimile: (212) 218-2200

#234901v3

CFM 2409 US
09/982,916



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年10月23日

出 願 番 号

Application Number:

特願2000-323040

出 願 人

Applicant(s):

キヤノン株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

RECEIVED

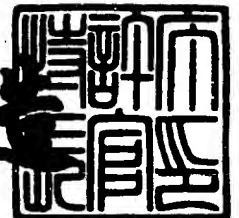
MAR 14 2002

Technology Center 2100

2001年11月16日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3100497

【書類名】 特許願

【整理番号】 4230019

【提出日】 平成12年10月23日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 7/00

【発明の名称】 フィルタ処理装置及びその方法

【請求項の数】 17

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 中山 忠義

【特許出願人】

 【識別番号】 000001007

 【氏名又は名称】 キヤノン株式会社

【代理人】

 【識別番号】 100076428

 【弁理士】

 【氏名又は名称】 大塚 康徳

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100101306

 【弁理士】

 【氏名又は名称】 丸山 幸雄

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100115071

 【弁理士】

 【氏名又は名称】 大塚 康弘

 【電話番号】 03-5276-3241

【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0001010

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 フィルタ処理装置及びその方法

【特許請求の範囲】

【請求項1】 入力されたデータに対して所定の係数を乗算する乗算手段と

、
前記乗算手段による乗算結果と前記入力データの一部を含む複数のデータの加算を行う加算手段と、

与えられたデータ群を格納すると共に、当該与えられたデータ群のうち、所望のデータを所定データ分遅延した遅延データを出力する格納手段と

を備える演算ユニットを複数備え、

前記複数の演算ユニットを用いて、外部入力データに対してフィルタ処理を行うことを特徴とするフィルタ処理装置。

【請求項2】 入力されたデータに対して所定の係数を乗算する乗算手段と

、
前記乗算手段による乗算結果と前記入力データの一部を含む複数のデータの加算を行う加算手段と

を備える複数の演算ユニットと、

各演算ユニットからのデータを入力、格納すると共に、当該データに対して遅延を行った遅延データを出力する格納手段とを備え、

外部入力データに対してフィルタ処理を行うことを特徴とするフィルタ処理装置。

【請求項3】 更に、前記演算ユニットに入力されるデータの切り替えを行う切り替え手段を備え、

前記切り替え手段は外部からの制御信号により、前記演算ユニットに入力されるデータの切り替えを行うことを特徴とする請求項1又は2に記載のフィルタ処理装置。

【請求項4】 前記切り替え手段の切り替えにより、前記演算ユニットは、順方向、逆方向のフィルタ処理の切り替えを行うことを特徴とする請求項3に記載のフィルタ処理装置。

【請求項 5】 前記外部からの制御信号は、前記所定の係数の正負の符号を制御することを特徴とする請求項 3 に記載のフィルタ処理装置。

【請求項 6】 更に、前記演算ユニットによるフィルタ処理結果に対してスケーリング処理を行うスケーリング手段を備えることを特徴とする請求項 1 又は 2 に記載のフィルタ処理装置。

【請求項 7】 前記入力されたデータは、前記外部入力データ、前記加算手段による加算結果、前記遅延データを含むことを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載のフィルタ処理装置。

【請求項 8】 前記複数のデータは更に、前記外部入力データ、前記遅延データを含むことを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載のフィルタ処理装置。

【請求項 9】 前記外部入力データは、画像を構成する各画素データであることを特徴とする請求項 1 乃至 8 のいずれか 1 項に記載のフィルタ装置。

【請求項 10】 前記フィルタ処理は離散ウェーブレット変換を含むことを特徴とする請求項 1 又は 2 に記載のフィルタ処理装置。

【請求項 11】 前記フィルタ処理は、L i f t i n g S c h e m e で示される方法に従うことを特徴とする請求項 10 に記載のフィルタ装置。

【請求項 12】 入力されたデータに対して所定の係数を乗算する乗算工程と、

前記乗算工程による乗算結果と前記入力データの一部を含む複数のデータの加算を行う加算工程と、

与えられたデータ群を所定の格納手段に格納すると共に、当該与えられたデータ群のうち、所望のデータを所定データ分遅延した遅延データを当該所定の格納手段から出力する格納工程と

を備える演算ユニットの制御方法を複数備え、

前記複数の演算ユニットの制御方法を用いて、外部入力データに対してフィルタ処理を行うことを特徴とするフィルタ処理装置の制御方法。

【請求項 13】 入力されたデータに対して所定の係数を乗算する乗算工程と、

前記乗算工程による乗算結果と前記入力データの一部を含む複数のデータの加算を行う加算工程と

を備える複数の演算ユニットの制御方法と、

各演算ユニットからのデータを所定の格納手段に入力、格納すると共に、当該データに対して遅延を行った遅延データを当該所定の格納手段から出力する格納工程とを備え、

外部入力データに対してフィルタ処理を行うことを特徴とするフィルタ処理装置の制御方法。

【請求項 1 4】 更に、前記演算ユニットに入力されるデータの切り替えを行う切り替え工程を備え、

前記切り替え工程は外部からの制御信号により、前記演算ユニットに入力されるデータの切り替えを行うことを特徴とする請求項 1 2 又は 1 3 に記載のフィルタ処理装置の制御方法。

【請求項 1 5】 更に、前記演算ユニットによるフィルタ処理結果に対してスケーリング処理を行うスケーリング工程を備えることを特徴とする請求項 1 2 又は 1 3 に記載のフィルタ処理装置の制御方法。

【請求項 1 6】 コンピュータに読み込ませることでフィルタ処理装置として機能するプログラムコードを格納する記憶媒体であって、

入力されたデータに対して所定の係数を乗算する乗算工程のプログラムコードと、

前記乗算工程による乗算結果と前記入力データの一部を含む複数のデータの加算を行う加算工程のプログラムコードと、

与えられたデータ群を所定の格納手段に格納すると共に、当該与えられたデータ群のうち、所望のデータを所定データ分遅延した遅延データを当該所定の格納手段から出力する格納工程のプログラムコードと

を備える、演算ユニットとして機能するプログラムコードを備え、

前記演算ユニットとして機能するプログラムコードを複数回用いて、外部入力データに対してフィルタ処理を行うことを特徴とする記憶媒体。

【請求項 1 7】 コンピュータに読み込ませることでフィルタ処理装置とし

て機能するプログラムコードを格納する記憶媒体であって、

入力されたデータに対して所定の係数を乗算する乗算工程のプログラムコードと、

前記乗算工程による乗算結果と前記入力データの一部を含む複数のデータの加算を行う加算工程のプログラムコードと

を備える、演算ユニットとして機能するプログラムコードと、

各演算ユニットからのデータを所定の格納手段に入力、格納すると共に、当該データに対して遅延を行った遅延データを所定の格納手段から出力する格納工程のプログラムコードとを備え、

外部入力データに対してフィルタ処理を行うことを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、外部入力データに対してフィルタ処理を行うフィルタ処理装置及びその方法に関するものである。

【0002】

【従来の技術】

画像、特に多値画像は非常に多くの情報を含んでおり、その画像を蓄積・伝送する際にはデータ量が膨大になってしまうという問題がある。このため画像の蓄積・伝送に際しては、画像の持つ冗長性を除く、或いは画質の劣化が視覚的に認識し難い程度で画像の内容を変更することによってデータ量を削減する高能率符号化が用いられる。

【0003】

例えば、静止画像の国際標準符号化方式としてISOとITU-Tにより勧告されたJPEGでは、画像データをブロックごと（8画素×8画素）に離散コサイン変換（DCT）でDCT係数に変換した後に、各係数を各々量子化し、さらにエントロピー符号化することにより画像データを圧縮している。ブロックごとにDCT、量子化を行なっているため、復号画像の各ブロックの境界で、所謂ブロック歪みが見える場合がある。

【 0 0 0 4 】

一方、新しい静止画像の国際標準符号化方式として J P E G 2 0 0 0 が検討されているが、J P E G 2 0 0 0 では、量子化の前に行う前処理の一つとして、ウェーブレット変換が提案されている。ウェーブレット変換は、現行 J P E G のようにブロック単位で処理を行うのではなく、入力データを連続的に処理するので、復号画像の劣化が視覚的に見えにくいという特徴がある。

【 0 0 0 5 】

図 1 は、変換用メモリ 1 0 1 と離散ウェーブレット変換部 1 0 2 の動作を説明する図である。

【 0 0 0 6 】

図 2 (a) は、離散ウェーブレット変換部 1 0 2 の基本構成を示すブロックである。同図左は、順方向の離散ウェーブレット変換（以下、DWT）を行う装置（離散ウェーブレット変換部 1 0 2 ）の基本構成であり、H 0 は低域通過特性を持つフィルタ、H 1 は高域通過特性を持つフィルタである。同図右は逆方向の離散ウェーブレット変換（逆離散ウェーブレット変換）を行う装置の基本構成を示す図である。フィルタの係数の一例を図 5 に示す。以下、説明を簡略化するため、同図に示す、 5×3 構成（低域 5 タップ・高域 3 タップ）の順方向のフィルタ係数を元に説明を行う。

【 0 0 0 7 】

図 2 (b) に示す入力画像が、左上から主走査方向に離散ウェーブレット変換部 1 0 2 に順次入力されてくる場合を例にとり説明する。画像の大きさは $N \times M$ とする。

【 0 0 0 8 】

図 2 (a) の左側より入力された画像データは、低域通過特性を持つフィルタ H 0 及び、高域通過特性を持つフィルタ H 1 によりフィルタリングされた後、その各々の結果が 2 : 1 にダウンサンプリングされ、最終的に入力と同数 ($N \times M$) のウェーブレット係数として出力される。

【 0 0 0 9 】

垂直方向に上述のフィルタリング処理をするために、画像データを変換用メモ

り 1 0 1 に格納し、垂直方向 M 画素に対して垂直フィルタリング処理を施しながら、水平方向にスキャンする。その結果、図 2 (c) に示すとおり、低域側のウェーブレット係数 L と高域側のウェーブレット係数 H との 2 つのサブバンドを生成する。

【 0 0 1 0 】

さらにサブバンドに分割し、水平方向のウェーブレット係数を得るために、ウェーブレット係数 L、H は変換用メモリ 1 0 1 にいったん全て格納される。

【 0 0 1 1 】

変換用メモリ 1 0 1 に格納されたウェーブレット係数は水平方向に読み出され、離散ウェーブレット変換部 1 0 2 により、水平方向 N 個の係数に対して H 0 及び H 1 によりフィルタリングを施し、結果は 2 : 1 にダウンサンプリングされる。図 2 (d) に示すように、係数 L に H 0 を施したものが L L、H 1 を施したものが L H であり、H に H 0 を施したものが H L、H 1 を施したものが H H である。L L、L H、H L、H H のそれぞれの大きさは $((N/2) \times (M/2))$ である。

【 0 0 1 2 】

以上説明した離散ウェーブレット変換方法とは異なる構成方法である、L i f t i n g S c h e m e と呼ばれる方法が知られている。図 3 に順方向の L i f t i n g S c h e m e、図 4 に逆方向の L i f t i n g S c h e m e の基本構成を示す。図中の p、u は L i f t i n g 係数と呼ばれるもので、図 6 に 5×3 フィルターと同一の出力を生成するための L i f t i n g 係数の例を示す。

【 0 0 1 3 】

図 6 に示す L i f t i n g 係数

$$p = (-1, -1) / 2$$

$$u = (1, 1) / 4$$

を元に、以下、図 3 に示した順方向の L i f t i n g S c h e m e の動作について説明する。

【 0 0 1 4 】

X は入力画像であり、同図に示すように (X 0, X 1, X 2, X 3, X 4, X

5. . .) である。入力画像は各々、偶数番目の画素、奇数番目の画素に分類される。入力画像のうち、偶数番目の画素を X_e 、奇数番目の画素を X_o とする。分類された画素は、*L i f t i n g*係数を乗ぜられた後に加算処理をなされ、低域側の係数と、高域側の係数に変換される。具体的に式で示すと、

$$\text{(高域側の係数)} \quad X'_o = X_o + p \cdot X_e$$

$$\text{(低域側の係数)} \quad X'_e = X_e + u \cdot X'_o$$

となる。なお、 X'_o 、 X'_e は、夫々低域側、高域側の係数である。また図中の k は、ウェーブレット係数を正規化するものであるが、ここで説明する内容の趣旨から外れるので、説明を省く。

【0015】

図4の逆方向の*L i f t i n g S c h e m e*の出力である画素の生成を具体的に式で示すと、

$$\text{(偶数番目の画素)} \quad X_e = X'_e - u \cdot X'_o$$

$$\text{(奇数番目の画素)} \quad X_o = X'_o - p \cdot X_e$$

となる。

【0016】

図3、4より、フィルタの構成が変われば、*L i f t i n g*係数、ならびに対象となる処理画素が異なることになるが、同様にして、順方向、逆方向の係数への変換を行うことができる。

【0017】

この*L i f t i n g S c h e m e*を用いると、量子化しない（あるいは量子化ステップ1で量子化する）場合、量子化後の情報のロスが無ければ圧縮符号化・復号化して復元されるデータが元のデータとまったく同じになる可逆的な変換を行なうことが出来る。J P E G 2 0 0 0では、*L i f t i n g S c h e m e*を採用して可逆的変換を実現している。

【0018】

*L i f t i n g S c h e m e*には、その他にフィルタ処理に要する演算量を少なくすることが出来るという特徴もあり、J P E G 2 0 0 0の 9×7 構成（低域9タップ・高域7タップ）のフィルタにも用いられている。

【 0 0 1 9 】

しかしながら、L i f t i n g S c h e m e を用いてフィルタ処理の演算量を少なくすることが出来るのは、フィルタの方向と処理のスキャン方向が同じ場合、すなわち、画像データを水平方向にスキャンしながら水平方向のフィルタ処理を行なうといった場合である。これは、前のサンプル点で高域と低域側の変換係数を出力するために演算した途中の結果を、次のサンプル点で再利用できるためである。

【 0 0 2 0 】

図 7 に示した L i f t i n g L a t t i c e を用いて、L i f t i n g S c h e m e における処理について説明する。

【 0 0 2 1 】

水平方向の画素の系列 $X_0, X_1, X_2, X_3, X_4, \dots$ があり、これらの画素の系列に対して水平方向の DWT 変換を行うと共に、右方向にスキャンする場合について考える。既に、黒丸で示した位置に対応する変換係数 s_4 と d_5 が求められているものとする。

【 0 0 2 2 】

s_4 は 9×7 構成 DWT フィルタの低域側の変換係数で、 d_5 は高域側の変換係数である。この s_4 と d_5 を求めるために、図 7 における灰色の丸で示した 8 つの変換データも既に計算されている。例えば、変換データの 1 つである d'_1 は以下のように計算される。

【 0 0 2 3 】

$$d'_1 = X_1 + \alpha \cdot (X_0 + X_2)$$

他の変換データも入力と乗算係数等が変わるだけで、演算式は同じである。ちなみに、J P E G 2 0 0 0 では以下のように係数が定義されている。

【 0 0 2 4 】

$$\begin{aligned} \alpha &= -1.586134342 \\ \beta &= -0.052980118 \\ \gamma &= 0.882911075 \\ \delta &= 0.443506852 \end{aligned}$$

同図において、灰色の丸が全部計算されている場合、次に求めるべき変換係数は s_6 と d_7 であり、先に計算した変換データや変換係数を再利用すれば、新たに計算しなければならないのは、変換データとして d'_9 と s'_8 の 2 つ、変換係数として s_6 と d_7 の 2 つの計 4 つだけで済む。1 変換係数当たりではわずか 2 つの計算で済むことになる。

【 0 0 2 5 】

1 つの計算の内訳は、3 つの入力の両端を足し合わせる加算 1 回、該加算結果に α 或いは β 、 γ 、 δ 等の係数を掛ける乗算 1 回、該乗算結果を真中の入力に加算する加算 1 回（2 回目の加算演算）である。以下ではこの計算を格子点演算と呼ぶことにする。

【 0 0 2 6 】

再利用する変換係数・データは、 d_5 、 s'_6 、 d'_7 の 3 つで、計算した値をレジスタに保持するだけで特別な制御も必要なく、簡単に再利用できることが図 7 の *L i f t i n g L a t t i c e* から容易に理解できる。

【 0 0 2 7 】

【発明が解決しようとする課題】

従来は、コーデックの一部の処理として、ウェーブレット変換等のフィルタ処理が必要な場合、順方向変換用のフィルタ処理部と逆方向変換用のフィルタ処理部といった 2 種類のフィルタ処理部を用意する必要があり、その結果回路規模が増大することとなった。また、フィルタが階層設計に適した構成になっておらず、回路の構造が複雑となり、開発やデバッグに要する時間もかかり、該機能を搭載した製品のコストアップの要因にもなっていた。

【 0 0 2 8 】

本発明は以上の問題点に対して鑑みてなされたものであり、乗算と加算を行う演算ユニットを複数用いて、フィルタ処理を行うことで、回路規模の増大を抑えると共に、回路の構造を簡単化することを目的とする。

【 0 0 2 9 】

【課題を解決するための手段】

本発明の目的を達成するために、例えば本発明のフィルタ処理装置は以下の構

成を備える。すなわち、

入力されたデータに対して所定の係数を乗算する乗算手段と、

前記乗算手段による乗算結果と前記入力データの一部を含む複数のデータの加算を行う加算手段と、

与えられたデータ群を格納すると共に、当該与えられたデータ群のうち、所望のデータを所定データ分遅延した遅延データを出力する格納手段と

を備える演算ユニットを複数備え、

前記複数の演算ユニットを用いて、外部入力データに対してフィルタ処理を行う。

【 0 0 3 0 】

本発明の目的を達成するために、例えば本発明のフィルタ処理装置は以下の構成を備える。すなわち、

入力されたデータに対して所定の係数を乗算する乗算手段と、

前記乗算手段による乗算結果と前記入力データの一部を含む複数のデータの加算を行う加算手段と

を備える複数の演算ユニットと、

各演算ユニットからのデータを入力、格納すると共に、当該データに対して遅延を行った遅延データを出力する格納手段とを備え、

外部入力データに対してフィルタ処理を行う。

【 0 0 3 1 】

【 発明の実施の形態 】

以下添付図面に従って、本発明を好適な実施形態に従って詳細に説明する。

【 0 0 3 2 】

〔 第 1 の実施形態 〕

前記従来例の説明では、水平方向の画素系列：X 0，X 1，X 2，X 3，X 4，．．．を図 7 のフィルタ処理への入力画素として説明したが、以下の実施形態では、図 8 に示すとおり、9 ライン分のデータのうち、縦一列の 9 画素のデータ（Y 0，Y 1，Y 2，Y 3，Y 4，．．．，Y 8）を入力するものとする。

【 0 0 3 3 】

以下では、垂直方向のフィルタ処理をしながら水平方向にスキャンをするといった処理についてまず考える。

【 0 0 3 4 】

垂直方向のフィルタ処理をしながら水平方向にスキャンをすると、9つの入力画素が次に処理する列の9画素に全面的に切り替わるため、1列前の変換係数を演算する時に計算した途中の演算結果を用いることができない。そのため、水平方向にスキャンして列が切り替わるごとに、図8における灰色の丸に対応する変換データをすべて計算する必要がある。黒丸に対応するのは変換係数（低域側の変換係数と高域側の変換係数）であるため、これは元々計算する必要がある。

【 0 0 3 5 】

よって、列が切り替わる度に10回の計算、1つの係数当たり5回の計算が必要になる。これは途中の計算結果を再利用できる場合の2.5倍の演算量になる。

【 0 0 3 6 】

この問題を解決するために、図9に示す構成を備えるフィルタ処理装置としての本実施形態における離散ウェーブレット変換を行う演算処理部について説明する。

【 0 0 3 7 】

図9において、

901、903、905は、ラインデータY8、Y9、Y10を入力する端子

911、913、915は、夫々のラインにおける変換係数もしくは変換データを格納し、与えられた変換係数もしくは変換データを遅延時間（遅延ライン分）遅延し、遅延時間分前のラインで、同じ列の変換係数もしくは変換データを出力するラインバッファ、

921、923、925、927は、演算された格子点データが得られる端子（格子点とも言う）を表わす。例えば、格子点921には以下の計算で求められる格子点データ d'_{921} が得られる。

【 0 0 3 8 】

$$d' 9 = Y 9 + \alpha \cdot (Y 8 + Y 1 0)$$

図 9 において、上記の式に基づいて計算された $d' 9$ をラインバッファ 9 1 1 に格納し、格納した $d' 9$ をラインバッファ 9 1 1 により 2 ライン分遅延し、2 ライン前の同じ列の変換データ $d' 7$ を得る。この $d' 7$ と $d' 9$ を用いて $s' 8$ を計算する。又、計算した変換データ $s' 8$ をラインバッファ 9 1 3 に格納する。以下、 $d 7$ 、 $s 6$ も同様にしてラインバッファ 9 1 3、9 1 5 を用いて求める。そして、同様に、求めた $d 7$ をラインバッファ 9 1 5 に格納する。

【 0 0 3 9 】

各ラインバッファ 9 1 1、9 1 3、9 1 5 は、水平方向にスキャンする長さに相当する容量を持ち、遅延時間は 2 ラインとなる。これは、同じ列のデータを用いた垂直方向フィルタ処理が、タイミング上 2 ライン毎に行なわれるからである。

【 0 0 4 0 】

もう少し詳しく説明すると、夫々のラインバッファから出力される変換係数 $d 5$ や変換データ $s' 6$ 、 $d' 7$ を求めるには、 $Y 0$ から $Y 8$ までの入力画素で計算できたが、変換係数 $s 6$ 、 $d 7$ を得るには、 $Y 1 0$ の入力を待たねばならないからである。

【 0 0 4 1 】

そして、次の垂直フィルタ処理のサイクルでは、水平方向に 1 列シフトして同様の計算が行なわれ計算結果が夫々のラインに応じたラインバッファに送られる。

【 0 0 4 2 】

このように、水平方向にスキャンしながら垂直方向フィルタ処理をして変換係数や変換データを次々とラインバッファに入力して蓄える。この時用いたラインデータ（入力画素） $Y 8$ と新たなラインデータ $Y 9$ 、 $Y 1 0$ を用いて次の水平スキャンを行なう。

【 0 0 4 3 】

この時、上記ラインデータに加え、3 つのラインバッファ 9 1 5、9 1 3、9 1 1 から出力される $d 5$ 、 $s' 6$ 、 $d' 7$ を用いることで、4 回の格子点演算を

行なうことで、2つの変換係数 s_6 、 d_7 を求めることができる。もちろん、次の水平スキャンに備え、上述の通り変換係数 d_7 と変換データ s'_8 、 d'_9 を夫々のラインバッファ 915、913、911 に再入力する。

【0044】

さらに次の水平スキャンでは、ラインデータ Y_{10} 、 Y_{11} 、 Y_{12} と、ラインバッファの出力 d_7 、 s'_8 、 d'_9 を用いることで2つの変換係数 s_8 、 d_9 を求めることができる。

【0045】

このようにして、垂直方向フィルタ処理をしながら水平方向へスキャンする場合にも、格子点演算2回につき1変換係数を求めることができる。

【0046】

以上説明した図9の構成はフィルタ処理後の変換係数を元の値に戻すための逆変換処理にも用いることができ、その場合の構成は図10に示すものとなる。これは、Lifting Lattice を用いたフィルタ処理の類似性から明らかなことなので、ここでの説明を省略する。

【0047】

上記図9の構成と同じ機能のものは、図11の構成でも実現できる。ラインデータ Y_8 を、新たに設けたラインバッファ 1101 に蓄えて、次の水平スキャンでは、新たなラインデータ Y_9 と Y_{10} のみを外部から入力し、既に入力済みのラインデータ Y_8 はラインバッファ 1101 から、 Y_{10} を2ライン分遅延させることで供給するようにしたものである。

【0048】

<変形例1>

本変形例1では、図12に示す格子点データ演算ユニットを図13のように複数接続することによって、フィルタ演算処理部を形成し、前記垂直方向のフィルタ処理を行なうものである。

【0049】

図12に示す格子点データ演算ユニットは、図11において4つの格子点各々に対応するデータを演算する演算部の内、1つの格子点のデータを演算する部分

と該演算に要するデータの入力元となるラインバッファ1つを取り出したものである。よって、演算機能等はすでに説明した内容と同様である。

【0050】

一方、図13における、1301、1303、1305、1307は、それぞれ図12に示した格子点データ演算ユニットであり、各ユニット毎に乗算係数が異なるが、基本的な構成は同じである。図13に示したフィルタ演算処理部の構成は、前記図11に示した演算処理部の構成を上記ユニット4つに置き換えただけであるため、機能的には図11とまったく同じである。

【0051】

逆変換用（逆方向のフィルタ処理）のフィルタ演算処理部も同一ユニットを用いて、図14のように構成できる。図13との違いは、各ユニット内の乗算係数が上下で入れ替わって正負の符号が反転していることである。

【0052】

本変形例で示した格子点データ演算ユニットを用いた上述のフィルタ演算処理部、及び逆変換のフィルタ演算処理部は、夫々のパラメータ（ α 、 β 、 γ 、 δ ）を調節した図12に示した格子点データ演算ユニットを用いて実現可能である。つまり、共通のハードウェア（ソフトウェアでも良いが）である格子点データ演算ユニットを用いることで、その両方のフィルタ処理（順方向、逆方向のフィルタ処理）が実現できる。

【0053】

また、上述の格子点データ演算ユニットは、遅延部がラインバッファに限定されるものではなく、 n 個のレジスタからなる遅延部であってもよい。

【0054】

一例として、図15に $n=2$ の場合を示す。

【0055】

一方、前記遅延部を格子点データ演算ユニット内に持たず、外部に共通にアクセスできるメモリを持ち、該メモリで遅延を実現することも可能である。その場合の格子点データ演算ユニットの構成を図16、図16に示した格子点データ演算ユニットを用いたフィルタ演算処理部の構成を図17に示す。

【 0 0 5 6 】

これ以降の説明に用いる格子点データ演算ユニットは、図の簡略化のため、該格子点データ演算ユニット内に遅延部を持っているものとして話を進めるが、遅延部を外部に持つ場合にも適用できることは、これまでの説明により明らかである。

【 0 0 5 7 】

また、各格子点データ演算ユニット内の乗算演算の係数は定数であるため、汎用の乗算器を用いる必要は無く、被乗数の足し合わせ方が決まっている定数乗算器を用いることができる。

【 0 0 5 8 】

また、本変形例で示した上述のフィルタ演算処理部の構成は、ウェーブレット変換などの特定のフィルタ処理に限定されるものではなく、一般のフィルタ処理にも適用可能であると同時に、以下の変形例においても同様であることは後述の説明により明らかである。

【 0 0 5 9 】

< 変形例 2 >

第 1 の実施形態における変形例 2 は、前述の各格子点データ演算ユニットの入力側に該ユニットへの入力を選択するセレクタを配し、該セレクタで選択するデータを順方向変換か逆方向変換かで切り替えることにより、共通のユニットを使用して、順方向と逆方向の両方の変換を実現するものである。

【 0 0 6 0 】

本変形例におけるフィルタ演算処理部の構成を図 1 8 に示す。図 1 8 において

1 8 0 0 は、変換の種類（順方向／逆方向）を指定する制御信号を入力する端子である。

【 0 0 6 1 】

1 8 0 1 ～ 1 8 0 4 は、それぞれパラメータ α 、 β 、 γ 、 δ を有し、定数乗算器と該乗算結果を加減算する機能を持つ格子点データ演算ユニットである。

【 0 0 6 2 】

1 8 1 1 ~ 1 8 1 4 は、端子 1 8 0 0 を介して入力される前記変換の種類を指定する制御信号に基づいて、出力を入力画素データか変換係数（もしくは係数データ）かを切り替える 4 入力 2 出力のセレクタである。

【 0 0 6 3 】

1 8 2 1、1 8 2 3 は、変換前の画像データを入力する端子、
1 8 2 5、1 8 2 7 は、変換後の係数データを入力する端子
1 8 3 1、1 8 3 3 は、順方向変換処理によって求められたデータ（変換係数）を出力する端子、

1 8 4 1、1 8 4 3 は、逆方向変換処理によって求められたデータ（入力画素データ）を出力する端子、
である。

【 0 0 6 4 】

端子 1 8 0 0 から入力される変換の種類を指定する制御信号に基づいて、各セレクタ 1 8 1 1 ~ 1 8 1 4 は選択出力するデータを切り替えると共に、各格子点データ演算ユニット 1 8 0 1 ~ 1 8 0 4 では順方向変換時には加算、逆方向変換時には減算を行なう。

【 0 0 6 5 】

そのため各格子点データ演算ユニット 1 8 0 1 ~ 1 8 0 4 は、定数を乗算した結果を加減算できるように図 1 9 に示す構成に変更した。回路構成上の実質的な違いは、加算器を加減算器 1 9 0 1 に置き換えたことである。

【 0 0 6 6 】

端子 1 8 0 0 に順方向変換を指定する制御信号が入力されると、各セレクタ 1 8 1 1 ~ 1 8 1 4 は左側の 2 入力（同図のセレクタ 1 8 1 1 では Y 9、Y 1 0）を選択して出力すると共に、各格子点データ演算ユニット 1 8 0 1 ~ 1 8 0 4 は、前記定数乗算結果を加算するモード（各格子点データ演算ユニットの反転回路 1 9 0 1 が加算モード）となり、図 1 3 と等価な構成となる。

【 0 0 6 7 】

一方、端子 1 8 0 0 に逆方向変換を指定する制御信号が入力されると、各セレクタ 1 8 1 1 ~ 1 8 1 4 は右側の 2 入力（同図では一段下の格子点データ演算ユ

ニットの2出力。ただし、セレクタ1814に関しては、s10, d11の2入力)を選択して出力すると共に、各格子点データ演算ユニット1811~1814は前記定数乗算結果を減算するモード(各格子点データ演算ユニットの加減算器1901は減算モード)となり、図14と等価な構成となる。

【0068】

図10からも分かるとおり、Y7は $C = -\alpha$ となるときユニット(1801)から、Y8は $C = -\beta$ となるときユニット(1802)から出力されるので、端子1841の方からY7が、端子1843の方からY8が出力されることになる。

【0069】

上記4入力2出力のセレクタ1811~1814では、順方向変換時と逆方向変換時では変換出力が別々の端子になってしまうが、セレクタ1812、1813を図20(a), (b)に示すクロススイッチ2001, 2003に置き換えると、図21に示すとおり、順方向変換時も逆方向変換時も同じ端子2101, 2103から変換出力を取り出せる。

【0070】

<変形例3>

本変形例のフィルタ演算処理部は、L i f t i n g S c h e m eによるフィルタ処理の最後に行なうスケーリングのための乗算処理を、順方向変換と逆方向変換とで同一の乗算器を用いて処理する構成に関する。

【0071】

スケーリングパラメータをKとすると、J P E G 2 0 0 0では、最終的な高域変換係数を得るために、前記L i f t i n g演算後の高域変換係数にKを乗算し、最終的な低域変換係数を得るために、前記L i f t i n g演算後の低域変換係数に $1/K$ を乗算する。

【0072】

図13に示したフィルタ演算処理部としての本変形例の垂直9/7-DWT演算処理部に、スケーリングのための乗算器(2201, 2203)を付加すると、図22に示す構成となる。同図において2201が高域変換データにKを乗算

する乗算器、2203が低域変換データに $1/K$ を乗算する乗算器である。

【0073】

また、図18に示した垂直9/7-DWT/I DWT演算処理部にスケーリングのための乗算器(2301, 2303、2311、2313)を付加すると、図23に示す構成となる。図23から分かるように、DWT演算スケーリング用に2個の乗算器2301、2303、I DWT演算スケーリング用に2個の乗算器2311、2313が必要である。

【0074】

同時に4つの乗算器を使うことは無く、ある時点で使用するのはいずれか一方の2個の乗算器だけである。

【0075】

本変形例は、前記変形例2の規則性をなるべく踏襲して、両変換モードで同じ2つの乗算器を使うようにした。

【0076】

図24に本変形例の垂直9/7-DWT/I DWT演算処理部の構成を示す。セレクタ2401を格子点データ演算ユニット1804の出力段に設け、該セレクタ2401の出力段に、共通に用いる2つの乗算器2411、2413を設けた。その他の構成並びに構成要素は前記変形例2で示した図18と同じである。

【0077】

<変形例4>

本変形例では、図9に示した演算処理部の変形例として、図25に示す演算処理部を示す。図9に示した演算処理部では、ラインバッファ915には d_7 を入力していたが、変形例では、 d_7 に予めパラメータ δ が乗算されている $\delta \cdot d_7$ を入力する。そして $\delta \cdot d_7$ を入力したラインバッファ915は同様にパラメータ δ が乗算されている出力値 $\delta \cdot d_5$ を出力する。それ以外の構成及び動作は、図9に示した演算処理部と同じである。

【0078】

この構成で、図9に示した演算処理部が行う演算の量に比べて演算量の増減はない。なお本変形例では d_7 を例として説明したが、これに限定されずに他の d

' 9 や s' 8 のうちのいくつか、もしくは全部であっても良い。その場合、 d' 9 を例に取り説明すると、ラインバッファ 9 1 1 には $\beta \cdot d'$ 9 が入力され、その出力は $\beta \cdot d'$ 7 で、 s' 8 の演算の際には、この $\beta \cdot d'$ 7 に β の乗算は行わない。

【 0 0 7 9 】

<変形例 5>

本変形例では、図 9 に示した演算処理部の変形例として、図 2 6 に示す演算処理部を示す。図 9 に示した演算処理部では、ラインバッファ 9 1 5 には d 7 を入力していたが、本変形例では $(\delta \cdot d$ 7 + s' 8) を入力し、更に、このラインバッファ 9 1 5 に入力する $(\delta \cdot d$ 7 + s' 8) を生成するために、 $\delta \cdot d$ 7 に、 s' 8 を加算する加算器 2 6 0 1 が備わっている。

【 0 0 8 0 】

図 2 6 では、加算器の数が増えてはいるが、変換係数 s 6 の演算に必要な加算処理が、例えば変形例 4 では 3 つの項の加算であったが、本変形例では 2 つの項の加算になっており、全体としての演算量は、例えば変形例 4 と同じである。

【 0 0 8 1 】

<変形例 6>

上述の変形例では、1 ライン前の同一列データをから計算した 3 つの変換データをそれぞれ 3 つの遅延部で遅延させたが、本変形例では、1 ライン前の同一列データをから計算した 1 つの変換係数と格子上の変換データを計算する途中の演算結果をそれぞれ第 1、第 2 の遅延部で遅延させ、新たな変換係数の計算に用いる。

【 0 0 8 2 】

本変形例の演算処理部の概略構成を図 2 7 に示す。図 9 に示した演算処理部において、遅延部 9 1 3、9 1 5 の 2 つを用いている。ラインバッファ 9 1 5 には、第 1 の実施形態と同様に、変換係数 d 7 を格納するが、ラインバッファ 9 1 3 には、 $\beta \cdot (d'$ 7 + d' 9) を格納し、この $\beta \cdot (d'$ 7 + d' 9) を計算するために必要となるラインデータ Y 6、 Y 7、 Y 8、 Y 9、 Y 1 0 は、図 2 7 の上部にある 5 つの端子から入力し、変換係数 d 7 を計算するのに必要なその他の

データ $\beta \cdot (d' 5 + d' 7)$ はラインバッファ 9 1 3 から与えられる。水平方向にスキャンしながら垂直フィルタ処理をするタイミング等は前記第 1 の実施形態とまったく同じであるため、これ以上の細かい説明は省略する。

【 0 0 8 3 】

本変形例では演算量が増えるが、遅延部の数は前記第 1 の実施形態より少なく済む。具体的には、1 係数当たり 3 回の格子点演算が必要になり（前記第 1 の実施形態では 2 回）、遅延部は第 1、第 2 の 2 つで済む。L i f t i n g S c h e m e を用いた変換では、逆方向の変換は格子点演算で用いる係数の順序と負号を逆に変換するだけで、まったく同様の構成で処理できる。すなわち、上述した各種実施形態を、図 2 8 に示す L i f t i n g L a t t i c e に適用した構成で、逆変換処理が行なえる。

【 0 0 8 4 】

〔第 2 の実施形態〕

第 1 の実施形態及びその変形例における離散ウェーブレット変換は、すべてハードウェアに関するもののみであったが、演算処理をそのまま数式化し、ラインバッファとして配列を確保することで、ほとんどそのままソフトウェア処理に応用できる。よって、ウェーブレット係数変換装置のみならずウェーブレット係数変換方式としてもよい。

【 0 0 8 5 】

以下、図 2 9 のフローチャートを用いて説明する。処理すべき画像データは不図示の入力デバイスから入力され、このフローチャートに従ったプログラムコードは不図示の CPU がアクセス可能なメモリ上に格納されているものとする。なお、以下で用いるインデックス n は $n > 1$ とする。

【 0 0 8 6 】

ステップ S 2 9 0 1 では、処理する画像データ 3 つ (Y_{n+2} , Y_{n+3} , Y_{n+4}) を不図示のメモリから読み出す。

【 0 0 8 7 】

ステップ S 2 9 0 3 では、3 つの格子点データ d'_{n+1} , S'_{n+1} , d_{n+1} の夫々を格納している、ラインバッファに相当する配列 H 1, H 2, H 3 から読み出す

【 0 0 8 8 】

ステップ S 2 9 0 5 では、 $d'_{n+3} = Y_{n+3} + \alpha \cdot (Y_{n+2} + Y_{n+4})$ を演算する

【 0 0 8 9 】

ステップ S 2 9 0 7 では、上記格子点データ d'_{n+3} を配列 H 1 に格納する。

【 0 0 9 0 】

ステップ S 2 9 0 9 は、 $S'_{n+2} + \beta \cdot (d'_{n+1} + d'_{n+3})$ を演算するステップ

ステップ S 2 9 1 1 では、上記格子点データ S'_{n+2} を配列 H 2 に格納する。

【 0 0 9 1 】

ステップ S 2 9 1 3 では、 $d'_{n+1} = d'_{n+1} + \gamma \cdot (S'_{n+2} + S'_n)$ を演算する

【 0 0 9 2 】

ステップ S 2 9 1 5 では、上記変換係数 d_{n+1} を配列 H 3 に格納するステップ

ステップ S 2 9 1 7 では、 $S_n = S'_n + \delta \cdot (d_{n-1} + d_{n+1})$ を演算する。

【 0 0 9 3 】

ステップ S 2 9 1 9 では、上記変換係数 S_n と d_{n+1} を次の処理ステージへ出力する。

【 0 0 9 4 】

各ステップの処理内容や全体の処理については、既に説明した実施形態から明白であるのでその説明は省略する。また、演算した格子点データや変換係数の格納先として、上記配列では無く、単なる変換あるいはレジスタ等も考えられる。

〔他の実施形態〕

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【 0 0 9 5 】

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体（または記録媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム(OS)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0096】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0097】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明した（図29に示す）フローチャートに対応するプログラムコードが格納されることになる。

【0098】

【発明の効果】

以上の説明により、本発明によれば、乗算と加算を行う演算ユニットを複数用いて、フィルタ処理を行うことで、回路規模の増大を押さえると共に、回路の構造を簡単化することができた。

【図面の簡単な説明】

【図 1】

従来例における変換用メモリ 1 0 1 と離散ウェーブレット変換部 1 0 2 の動作を説明する図である。

【図 2】

(a) は、離散ウェーブレット変換部 1 0 2 の基本構成を示すブロック図、(b) は入力画像を示す図、(c) は生成された L サブバンド、H サブバンドを示す図、(d) は HH、HL、LH、LL サブバンドを示す図である。

【図 3】

順方向の L i f t i n g S c h e m e の基本構成を示す図である。

【図 4】

逆方向の L i f t i n g S c h e m e の基本構成を示す図である。

【図 5】

フィルタ係数を示す図である。

【図 6】

L i f t i n g 係数を示す図である。

【図 7】

L i f t i n g L a t t i c e の構成を示す図である。

【図 8】

L i f t i n g L a t t i c e の構成を示す図である。

【図 9】

本発明の第 1 の実施形態における順方向の演算ユニットの構成を示す図である。

【図 1 0】

本発明の第 1 の実施形態における逆方向の演算ユニットの構成を示す図である。

【図 1 1】

図 9 に示された演算ユニットと同じ機能を有し、他の構成をとる演算ユニットの構成を示す図である。

【図 1 2】

本発明の第 1 の実施形態における変形例 1 で用いる格子点データ演算ユニットの構成を示す図である。

【図 1 3】

図 1 2 に示したユニットを複数接続することで形成されるフィルタ演算処理部の構成を示す図である。

【図 1 4】

本発明の第 1 の実施形態における変形例 1 で用いる逆変換用のフィルタ演算処理部の構成を示す図である。

【図 1 5】

格子点データ演算ユニットを n 個のレジスタからなる遅延部で構成した場合、一例として、 $n = 2$ とした場合の構成を示す。

【図 1 6】

外部に共通にアクセスできるメモリを持ち、該メモリで遅延を実現する場合の格子点データ演算ユニットの構成を示す図である。

【図 1 7】

図 1 6 に示した格子点データ演算ユニットを用いたフィルタ演算処理部の全体の構成を示す図である。

【図 1 8】

本発明の第 1 の実施形態における変形例 2 におけるフィルタ演算処理部の構成を示す図である。

【図 1 9】

図 1 8 に示した格子点データ演算ユニットを変更した場合の構成を示す図である。

【図 2 0】

クロススイッチを示す図である。

【図 2 1】

本発明の第 1 の実施形態における変形例 2 のフィルタ演算処理部の構成を示す図である。

【図 2 2】

図 1 3 に示した垂直 9 / 7 - DWT 演算処理部にスケーリングのための乗算器を付加した構成を示す図である。

【図 2 3】

図 1 8 に示した垂直 9 / 7 - DWT / I DWT 演算処理部にスケーリングのための乗算器を付加した構成を示す図である。

【図 2 4】

本発明の第 1 の実施形態における変形例 3 の垂直 9 / 7 - DWT / I DWT 演算処理部の構成を示す図である。

【図 2 5】

本発明の第 1 の実施形態における変形例 4 の演算ユニットの構成を示す図である。

【図 2 6】

本発明の第 1 の実施形態における変形例 5 の演算ユニットの構成を示す図である。

【図 2 7】

本発明の第 1 の実施形態における変形例 6 の演算ユニットの構成を示す図である。

【図 2 8】

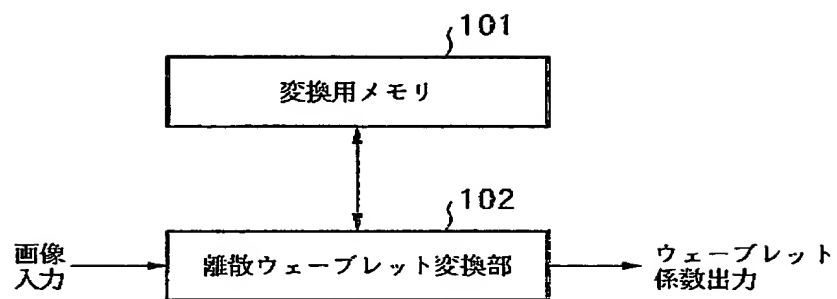
逆変換の L i f t i n g L a t t i c e を示す図である。

【図 2 9】

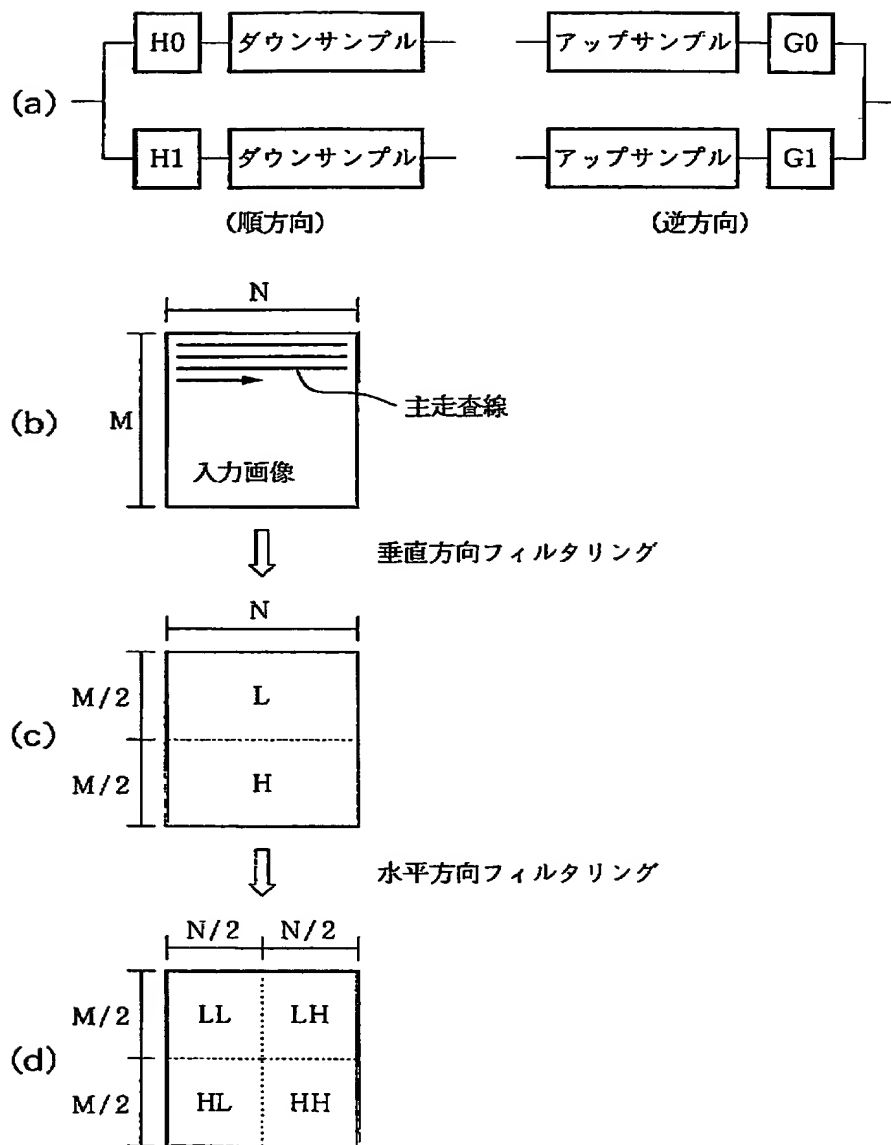
本発明の第 2 の実施形態である離散ウェーブレット変換処理方式のフローチャートである。

【書類名】 図面

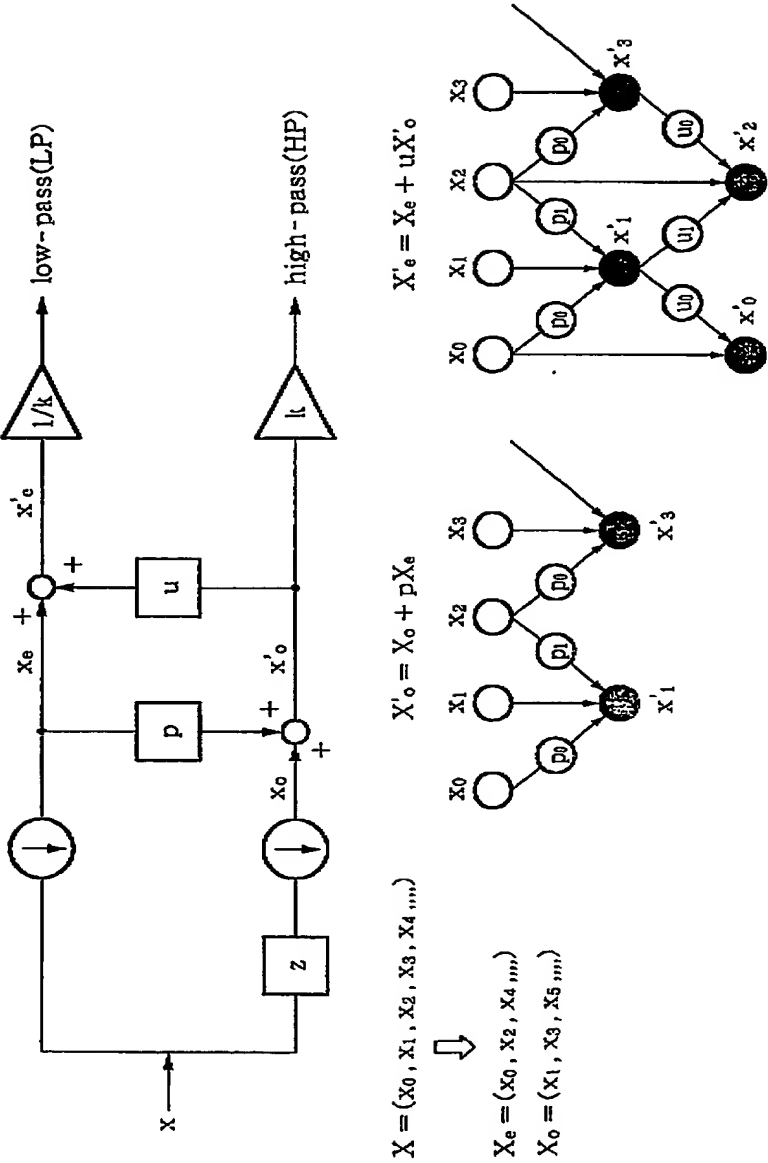
【図 1】



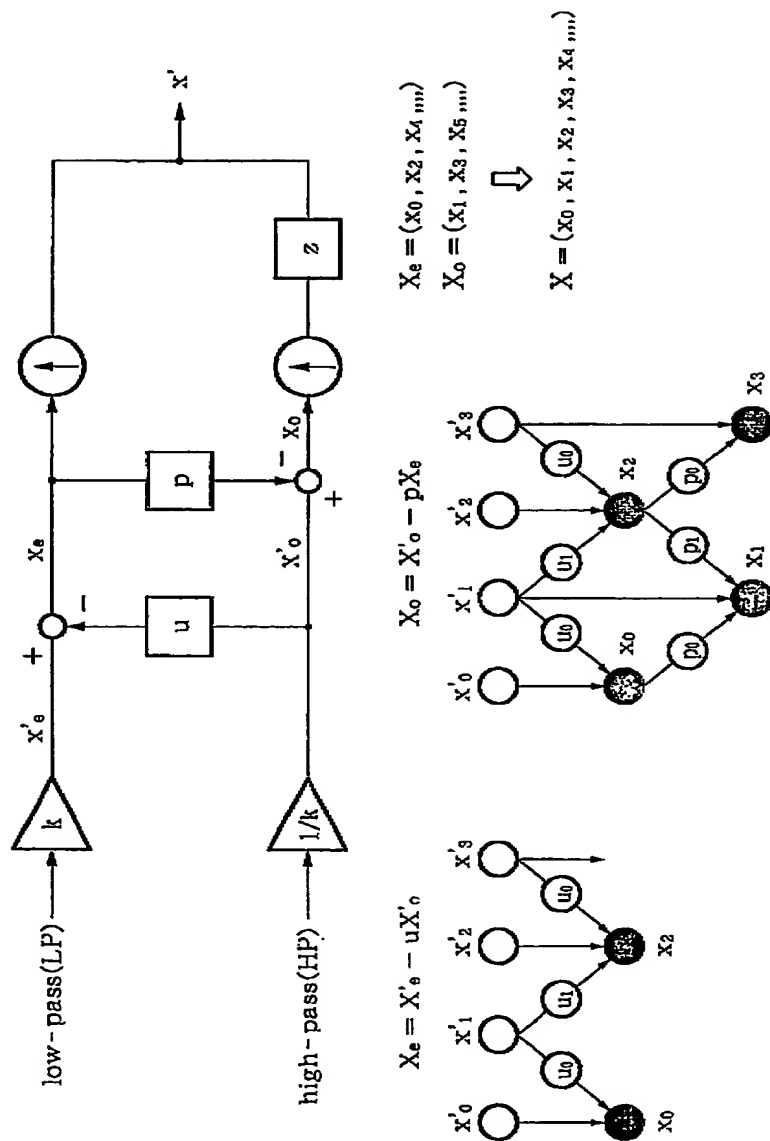
【図 2】



【図 3】



【図 4】



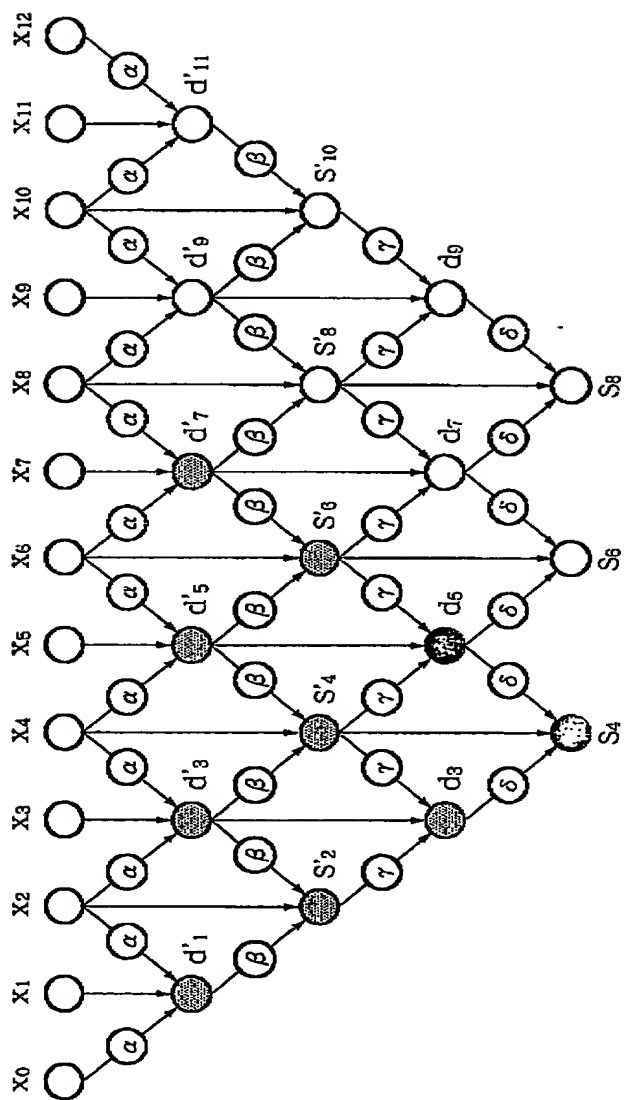
【図 5】

| Filter | Filter coefficients | |
|--------------|--|--|
| | 順方向 | 逆方向 |
| 5 × 3 filter | $H0 = [-1, 2, 6, 2, -1]/8$ $H1 = [-1, 2, -1]/2$ | $G0 = [1, 2, 1]/2$ $G1 = [-1, -2, 6, -2, -1]/8$ |

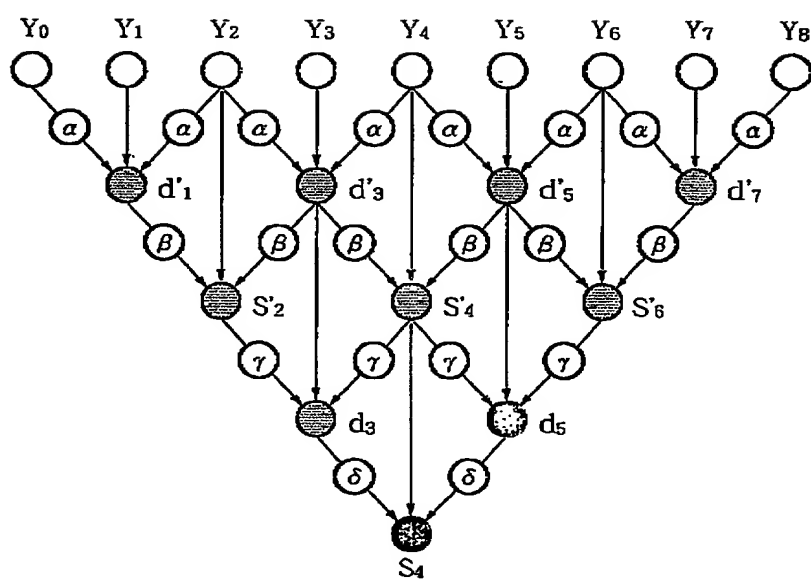
【図 6】

| Filter | Lifting coefficients | |
|--------------|------------------------------------|------------------------------------|
| | 順方向 | 逆方向 |
| 5 × 3 filter | $p = (-1, -1)/2$ $u = (1, 1)/4$ | $p = (-1, -1)/2$ $u = (1, 1)/4$ |

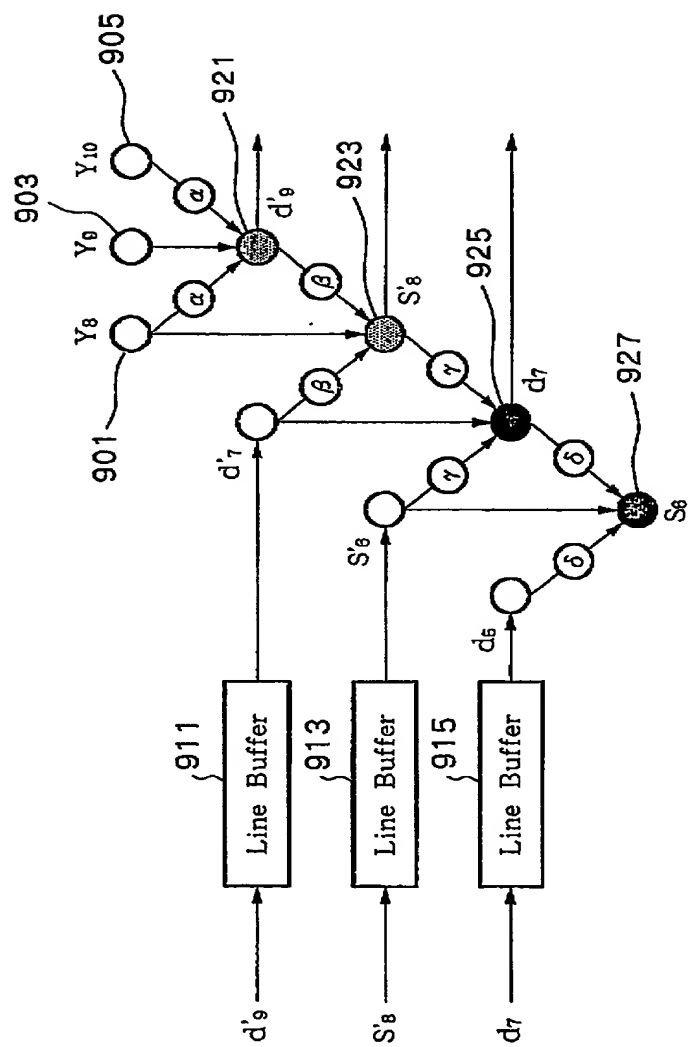
【図 7】



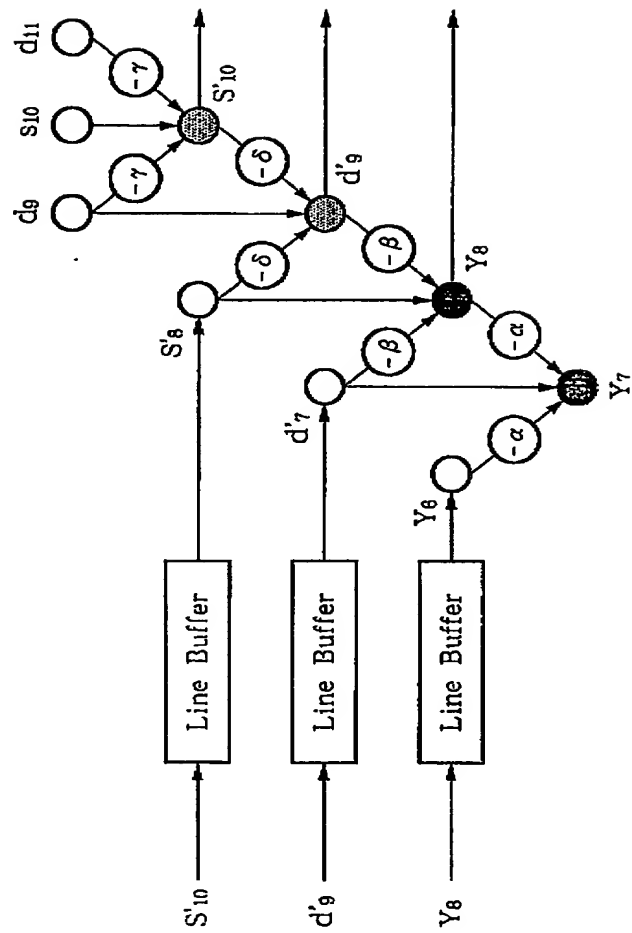
【図 8】



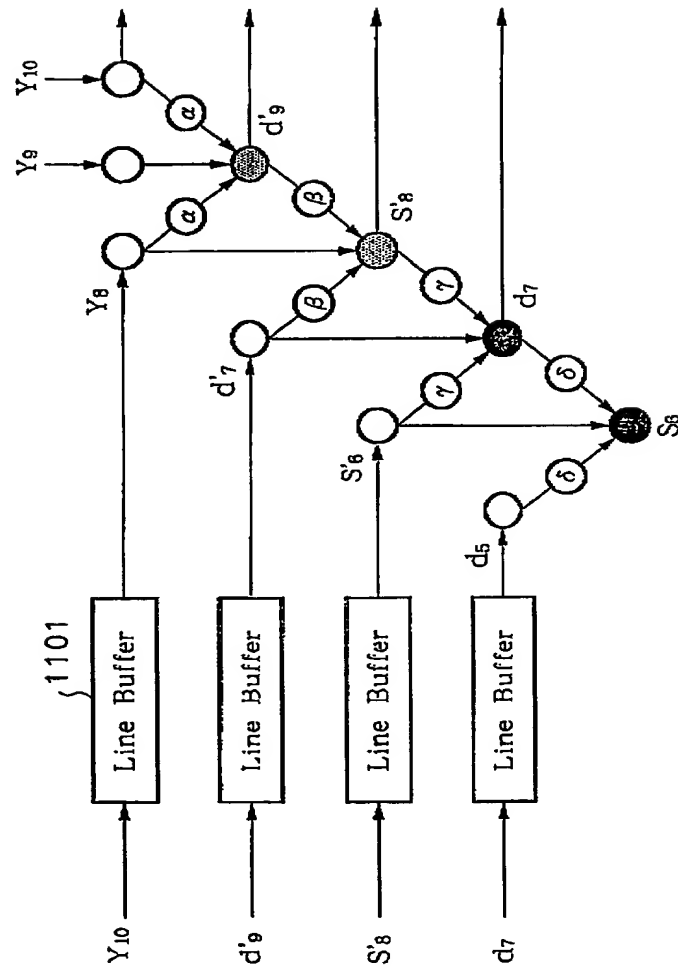
【図9】



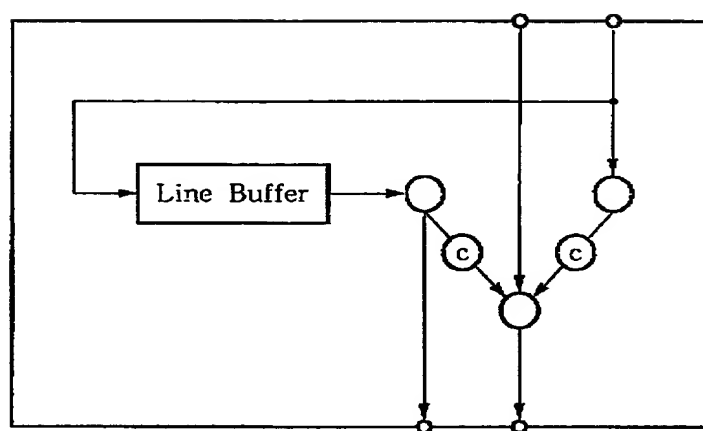
【図 1 0】



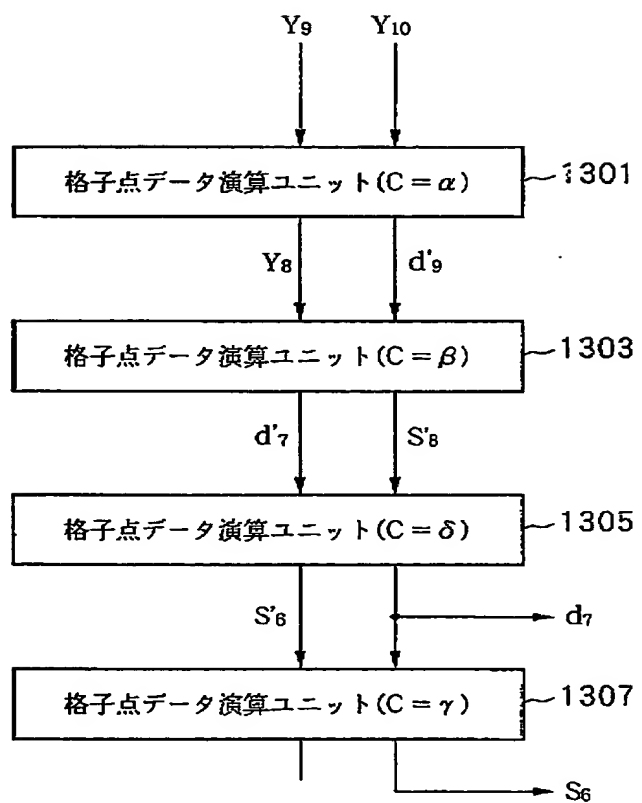
【図 11】



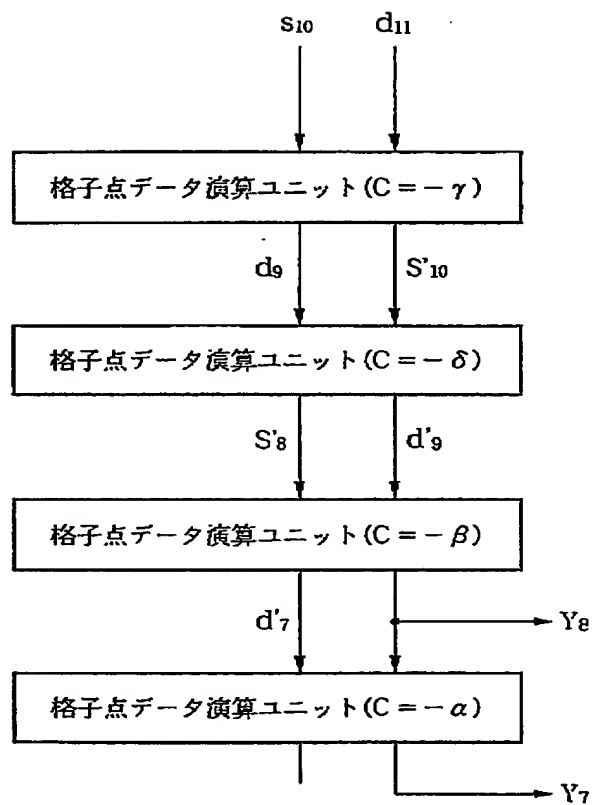
【図 1 2】



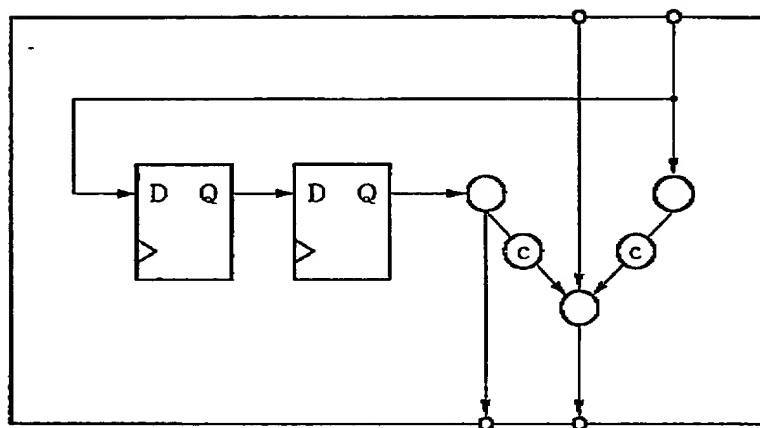
【図 1 3】



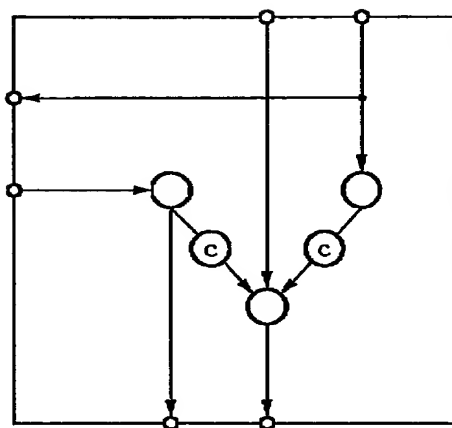
【図 1 4】



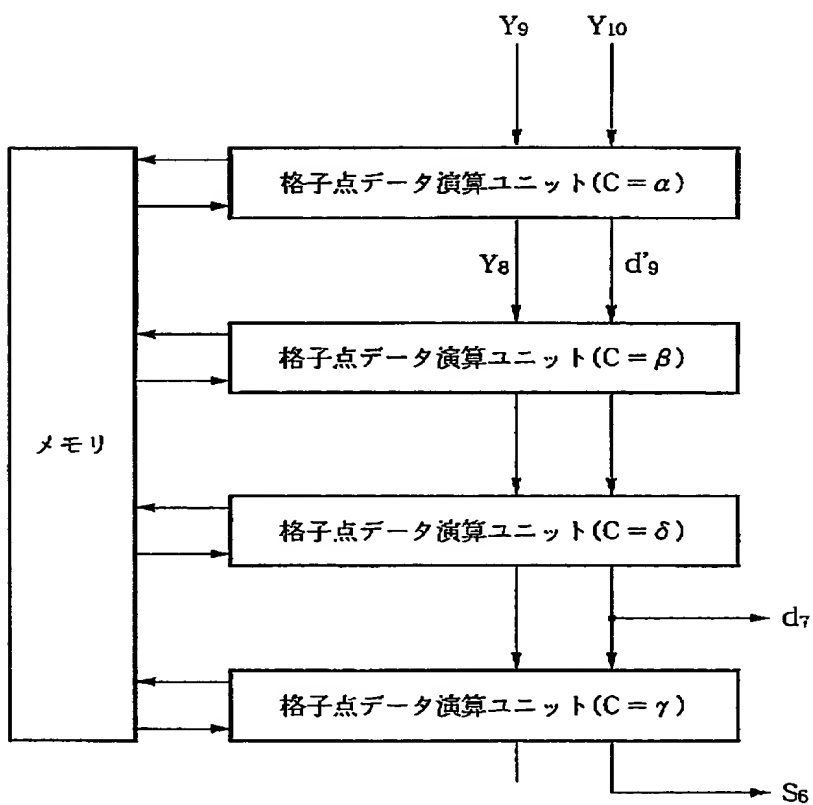
【図 1 5】



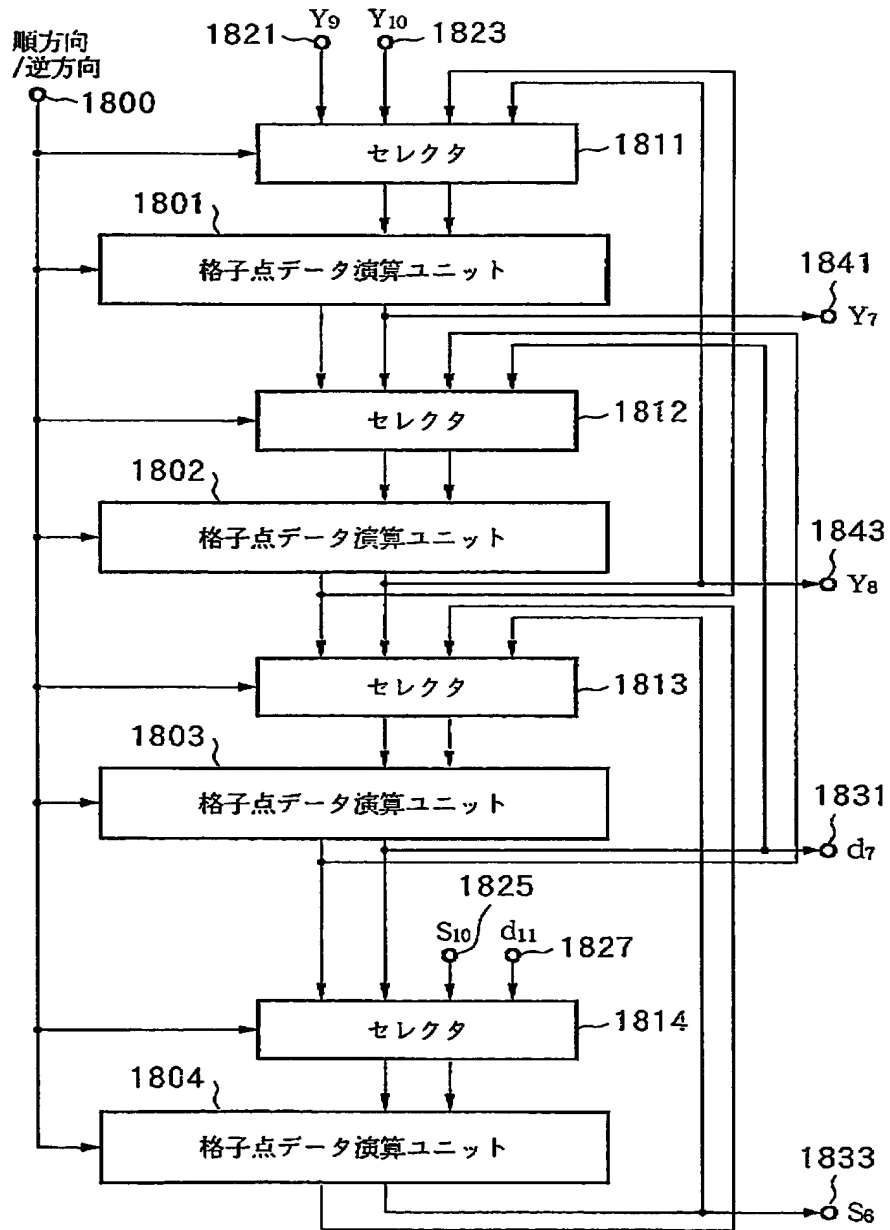
【図 16】



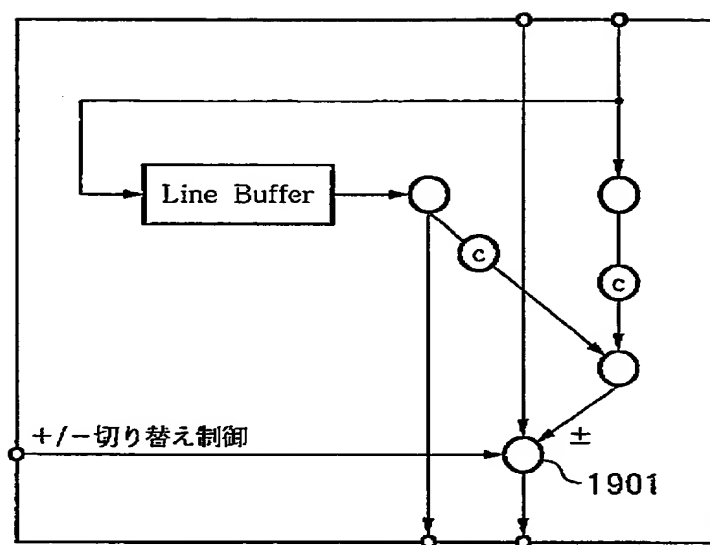
【図 17】



【図 1 8】

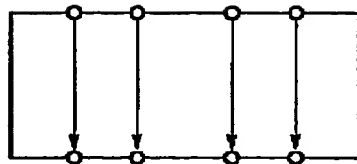


【図 1 9】

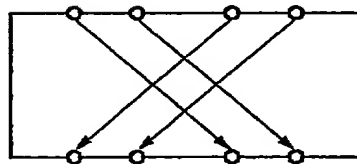


【図 2 0】

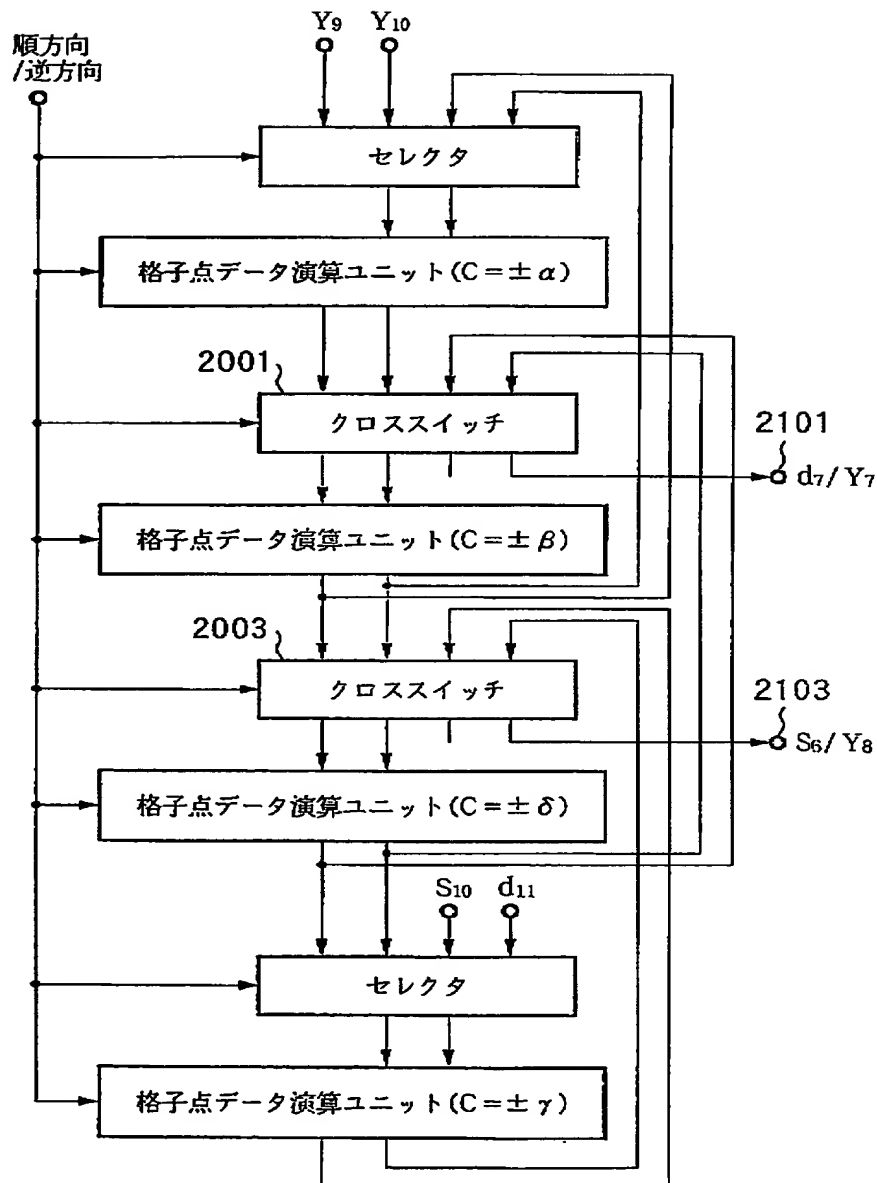
(a)



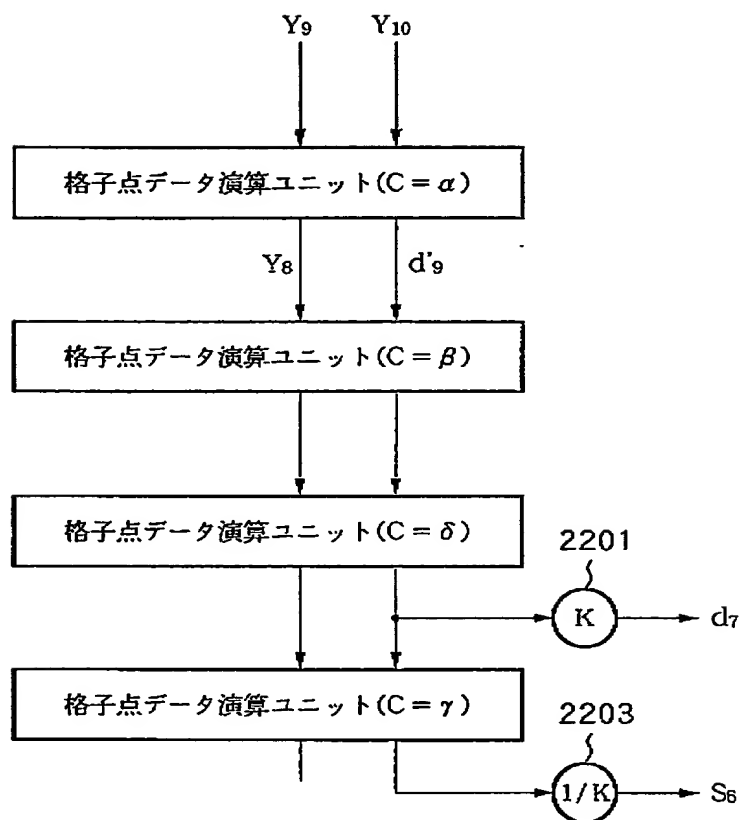
(b)



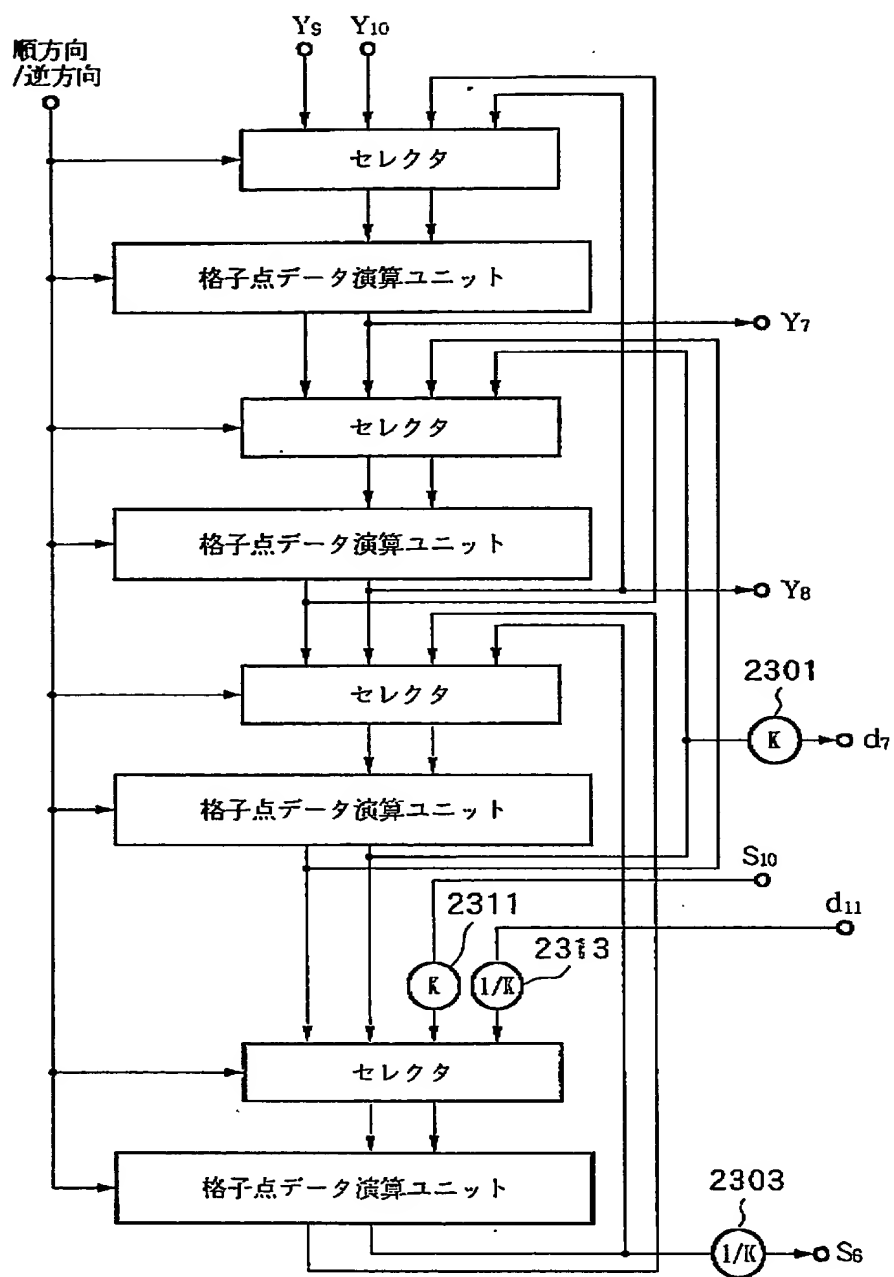
【図 21】



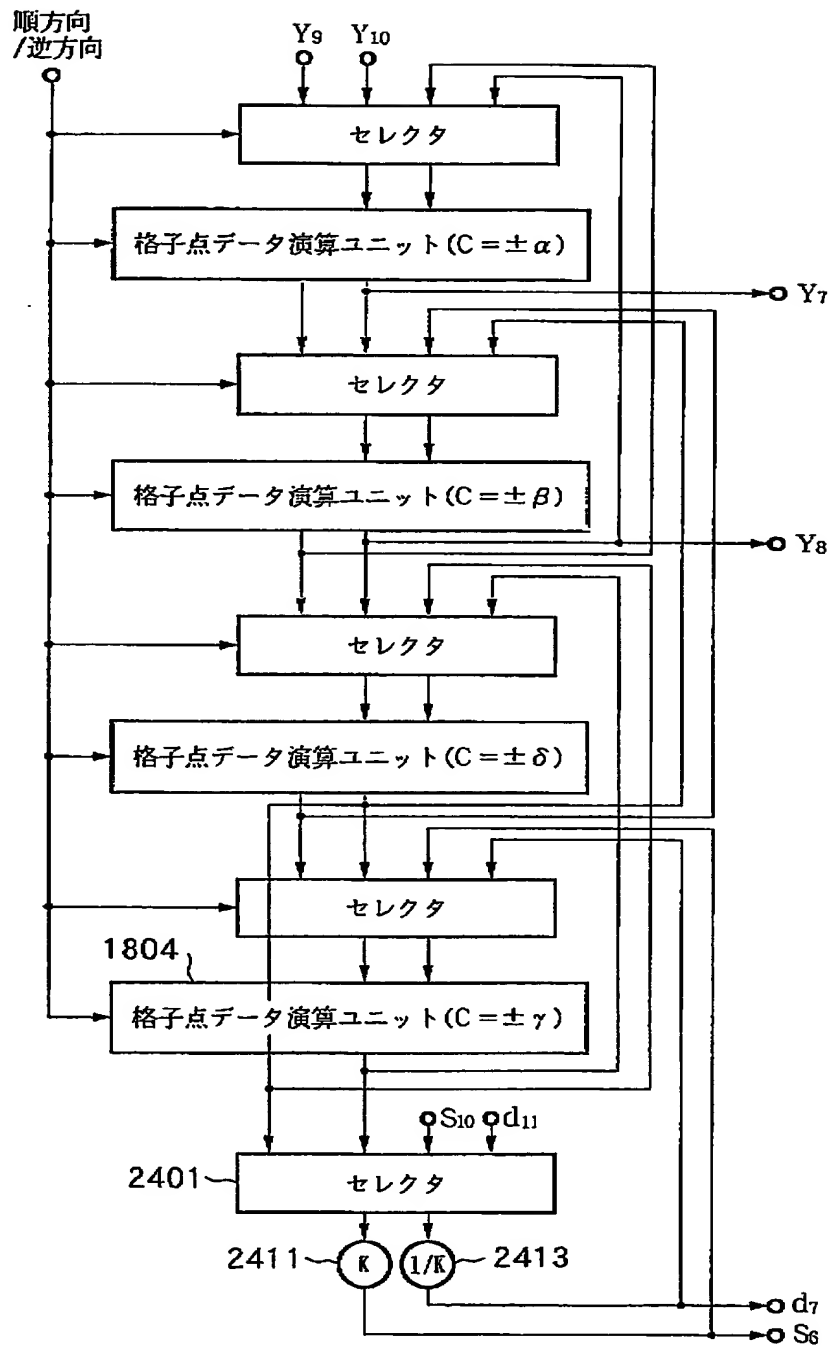
【図 2 2】



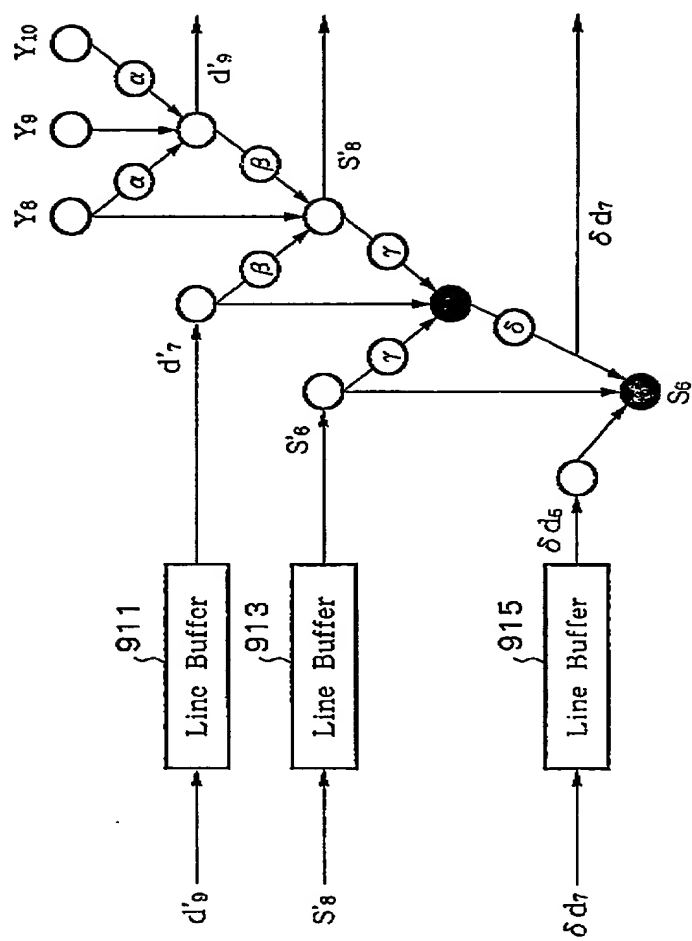
【図 23】



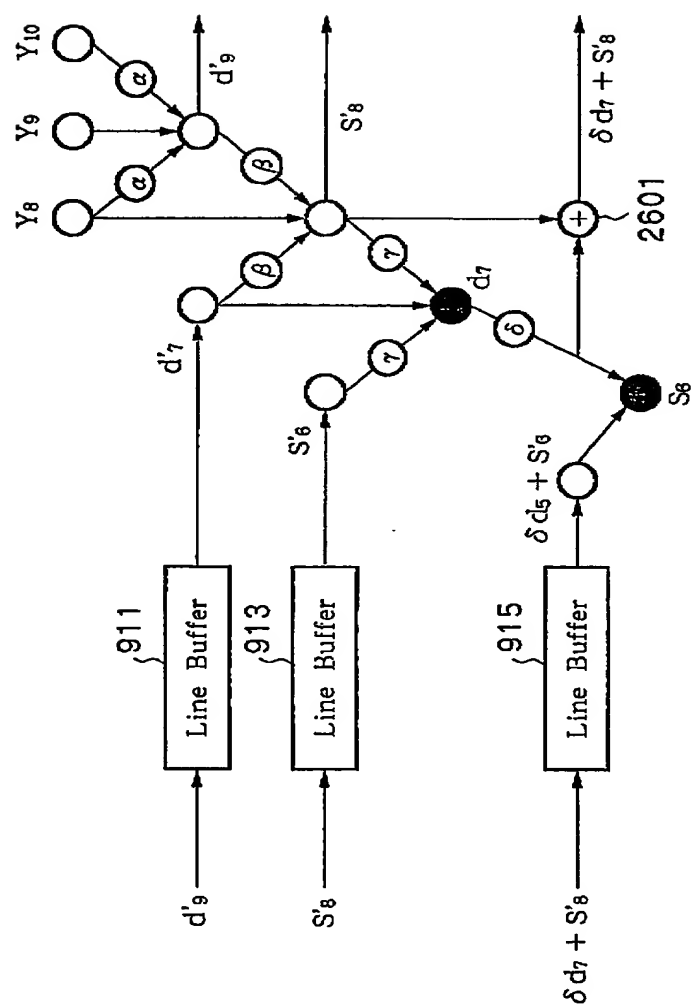
【図 24】



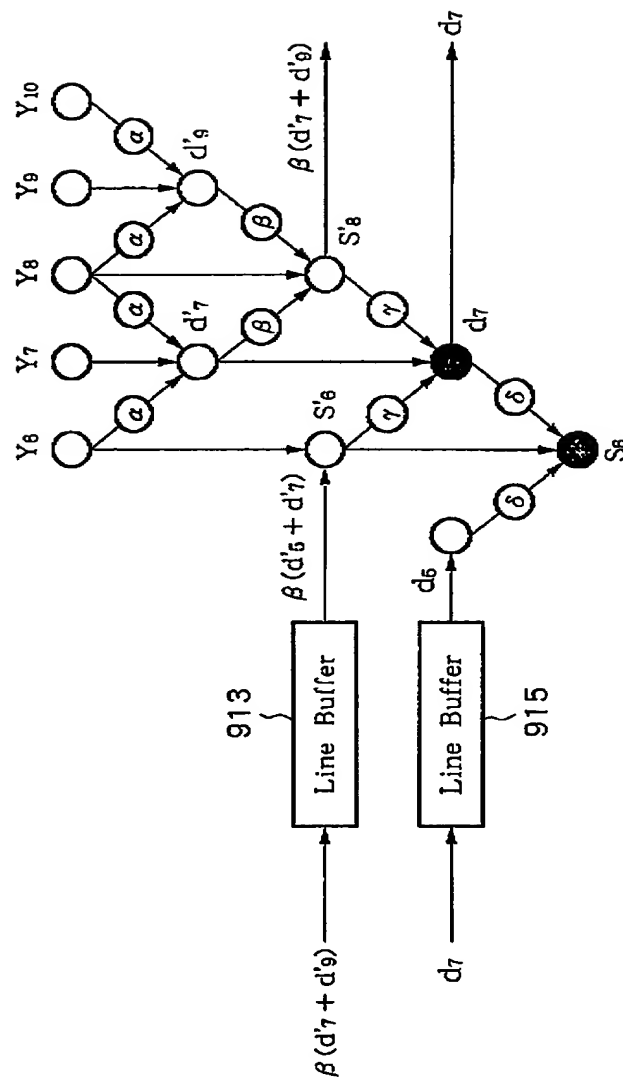
【図 2 5】



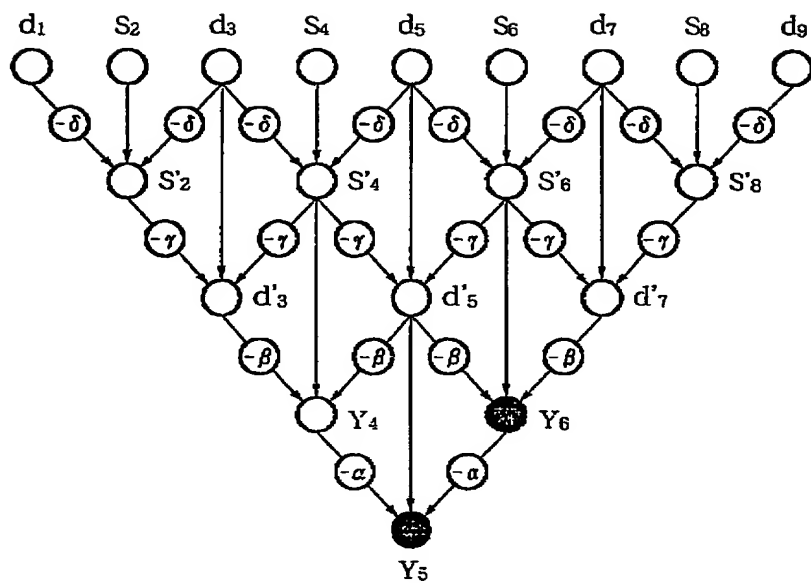
【図 2 6】



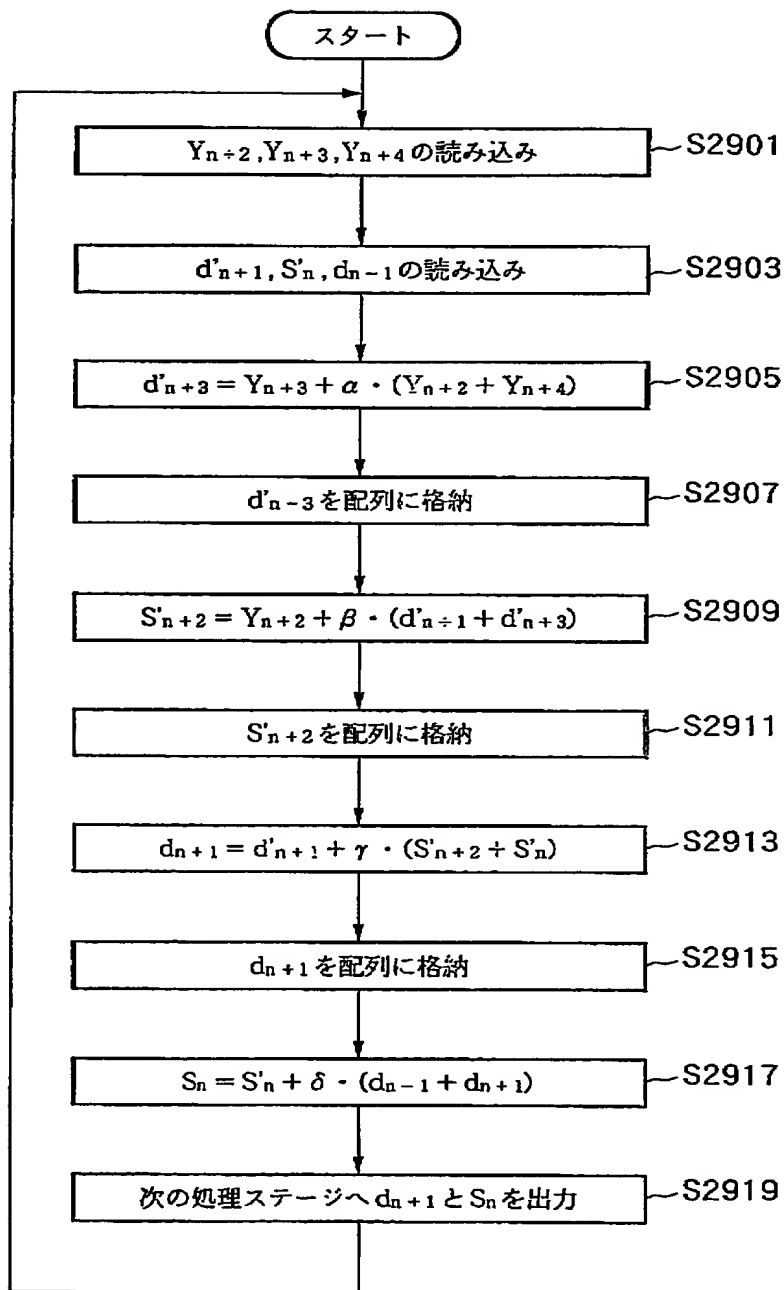
【図 2 7】



【図 2 8】



【図 2 9】



【書類名】 要約書

【要約】

【課題】 乗算と加算を行う演算ユニットを複数用いて、フィルタ処理を行うことで、回路規模の増大を押さえると共に、回路の構造を簡単化すること。

【解決手段】 処理する画像データ Y_{n+2} , Y_{n+3} , Y_{n+4} を読み出し (S 2 9 0 1)、3つの格子点データ d'_{n+1} , S'_n , d_{n-1} の夫々を格納する、ラインバッファに相当する配列 H_1 , H_2 , H_3 から読み出す (S 2 9 0 3)。そして、 $d'_{n+3} = Y_{n+3} + \alpha \cdot (Y_{n+2} + Y_{n+4})$ を演算し (S 2 9 0 5)、 d'_{n+3} を配列 H_1 に格納する (S 2 9 0 7)。次に、 $S'_{n+2} + \beta \cdot (d'_{n+1} + d'_{n+3})$ を演算し (S 2 9 0 9)、 S'_{n+2} を配列 H_2 に格納する (S 2 9 1 1)。次に、 $d'_{n+1} = d'_{n+1} + \gamma \cdot (S'_{n+2} + S'_n)$ を演算し (S 2 9 1 3)、 d_{n+1} を配列 H_3 に格納する (S 2 9 1 5)。次に、 $S_n = S'_n + \delta \cdot (d_{n-1} + d_{n+1})$ を演算し (S 2 9 1 7)、 S_n と d_{n+1} を次の処理ステージへ出力する (S 2 9 1 9)。

【選択図】 図 2 9

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日
[変更理由] 新規登録
住 所 東京都大田区下丸子3丁目30番2号
氏 名 キヤノン株式会社